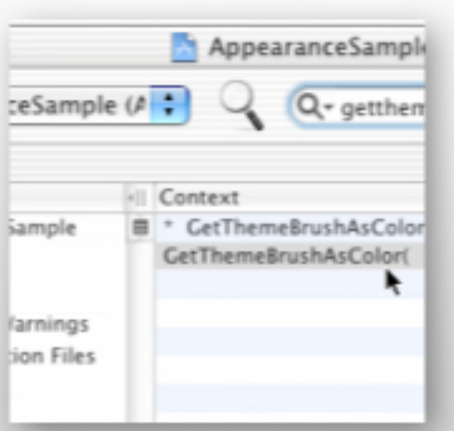
Increasing Productivity with Apple Tools

Session 303



Increasing Productivity with Apple Tools

Anders Bertelrud
Xcode Lead Engineer

What We'll Cover

- Moving to Xcode
- Optimizing build performance
- Building for multiple OS releases
- Workflow and automation
  - Source code management
  - Accessing documentation
  - Scripting and automation

The Common Theme

Moving to Xcode

Moving to Xcode

- It's easy to get started
- Many migration paths
  - Create project, add sources
  - Open Project Builder project
  - Import CodeWarrior project
  - Wrap Makefile in Xcode project
- Convert targets at your convenience
  - Native Xcode targets provide additional features

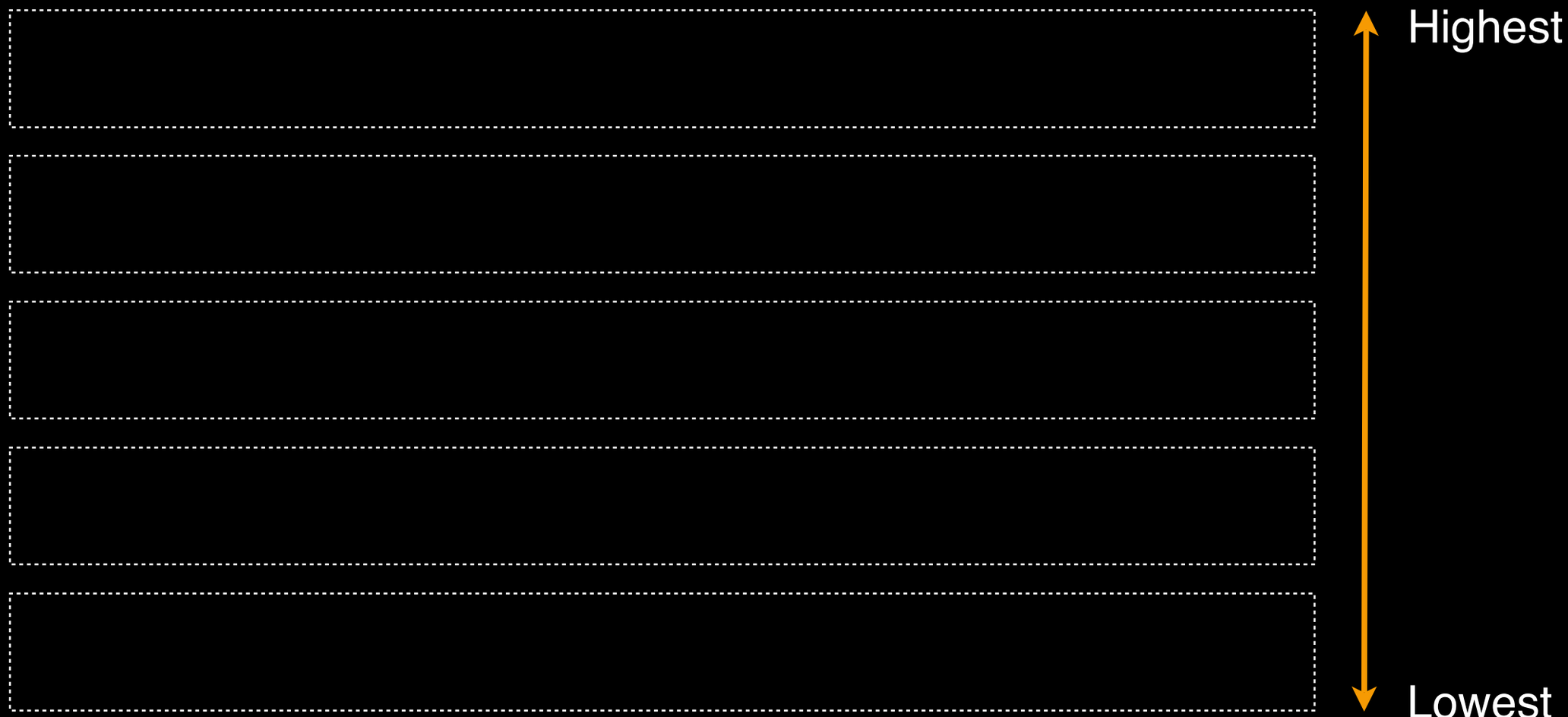
Target Types

- Native Xcode targets
  - Applications, Frameworks, Libraries, Plug-ins
- Project Builder targets
  - All product types supported by Project Builder
- External targets
  - Invoke any external build tool (e.g. Make or Ant)

Build Settings

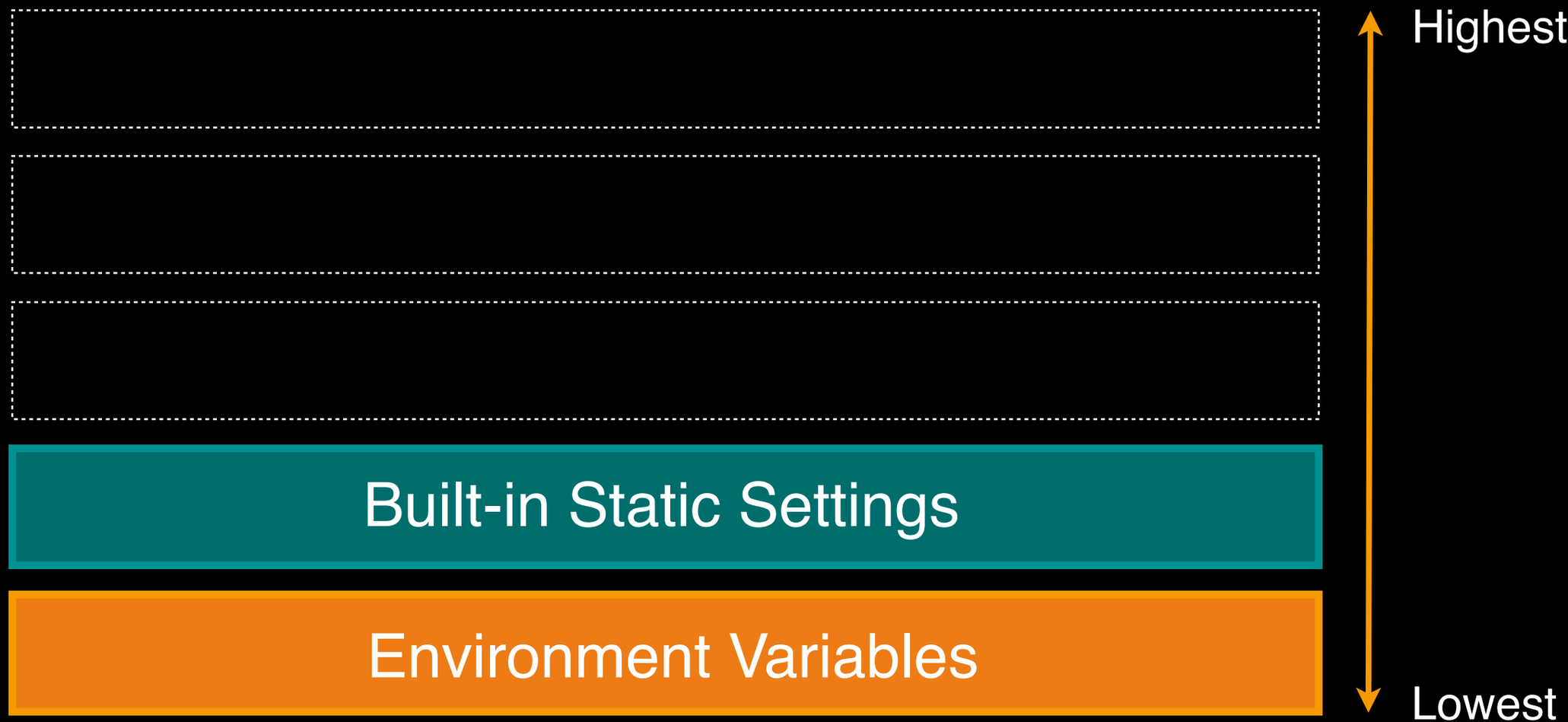
- Build settings control the inner workings of a build
- Usually set on a particular target
- Can also set in a build style to provide variation
  - e.g., debug vs. release builds
- Settings in build styles override lower levels
- Passed as environment variables for tools

Precedence of Build Settings

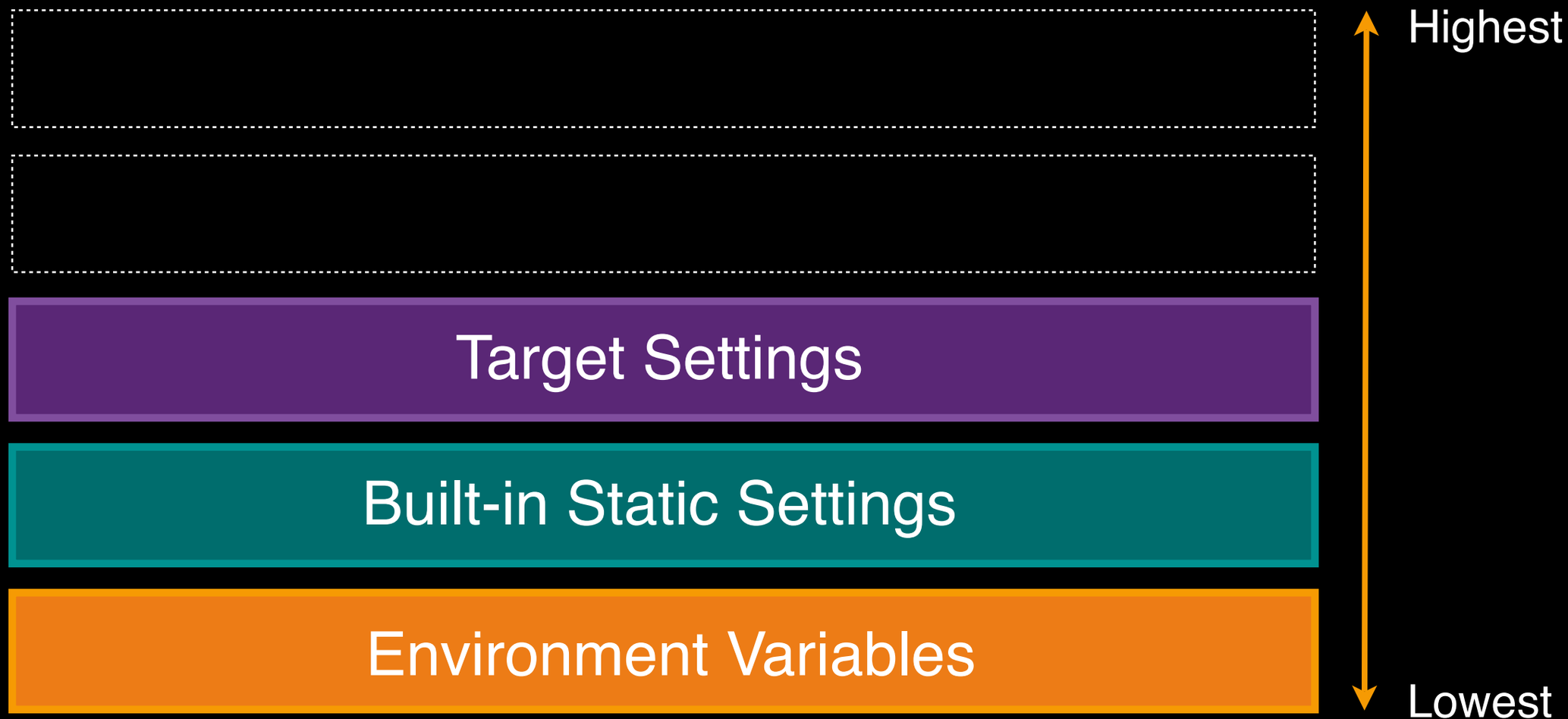


Precedence of Build Settings

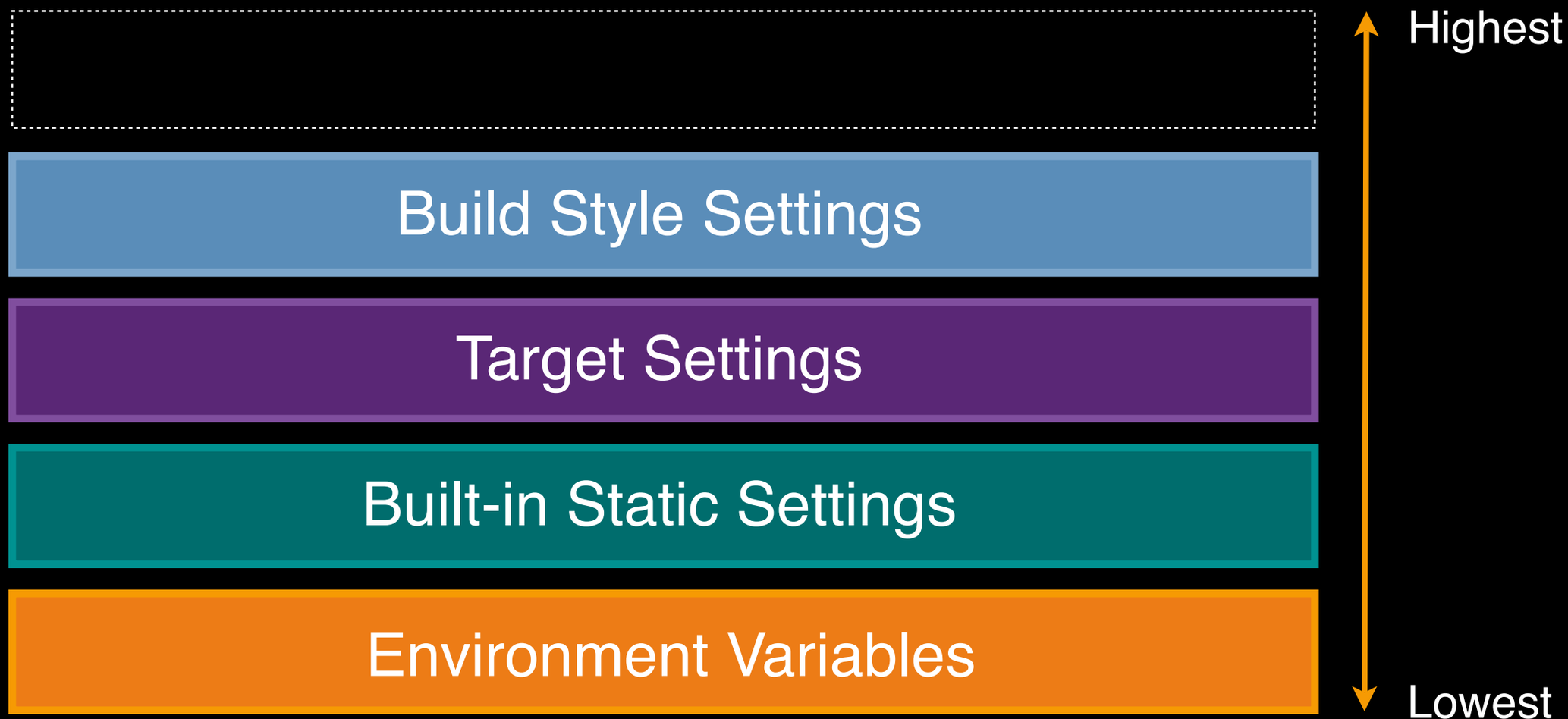
Precedence of Build Settings



Precedence of Build Settings



Precedence of Build Settings



Precedence of Build Settings

Build Styles

- Dynamically override target settings
- Avoid having to duplicate targets

Build Styles

- Dynamically override target settings
- Avoid having to duplicate targets
- Examples:
  - Debugging symbols on or off
  - Zero Link enabled or disabled
  - Different optimization levels
  - Alternate algorithms
  - Features enabled or disabled

Build Rules

- Rules define how source files are processed
- First matching rule is the active one
- Each rule can match:
  - Category of file types (e.g., “C++ source files”)
  - Particular filename pattern (e.g., “*.[hc]”)
- Use rules to select compiler and version
  - e.g., GCC 3.1 vs 3.3, or Javac vs Jikes

Optimizing Build Performance

Optimizing Build Performance

- Fast, powerful build system
  - No dependency on external builds tools such as *Jam*
  - Tightly integrated with the user interface
- There are still opportunities for project-specific tuning
  - Precompiled headers (PCH)
  - Parallel and distributed builds
  - Zero Link
  - Fix and Continue

Precompiled Headers

- GCC 3.3 supports PCH (precompiled headers)
- Easy to set up using Xcode's target inspector
  - Create a prefix header to include common API
  - Enable the "precompile prefix header" setting
- Keep in mind:
  - Prefix header gets included for all source files in target
  - Only put infrequently changed headers in the PCH

Distributed Builds

- Xcode takes advantage of multiple processors
  - But why limit yourself to the ones in your own machine?
- Compiles can be distributed to other machines
- “Free love” vs. restricted mode
  - Can either use Rendezvous or specify a list of servers
- Requires GCC 3.3 and Xcode targets

Zero Link

- Link time is a significant part of turnaround time
- Typical editing affects few source files at a time
- Zero Link eliminates the link step by running directly from the object files
- Enabled for *Development* builds only, by default—*Deployment* builds generate full binaries

Fix and Continue

- Fix while debugging
- Enabled by default
- Great for localized changes
  - Logic errors, off-by-one errors, etc.
  - Edit within blocks

Demo

Scott Tooker
Xcode Engineer

Building for Multiple OS Releases

Building for Multiple OS Releases

- Run on one, develop for all
  - e.g., run Xcode on Jaguar, build for Panther Preview
- Each target can specify OS release to build against
- Xcode Developer Preview supports
  - 10.1.5
  - 10.2.6
  - Panther Preview

Multiple OS Release Support

- APIs for each OS release are packaged in an “SDK”
- What’s in an SDK?
  - Headers
  - Stub libraries
- SDKs can be added and removed independently
- Take advantage of OS features when available

Source Tree Support

- Access external source bases
  - PowerPlant, Qt, etc.
- Absolute or relative paths
- Used by CodeWarrior importer

Workflow and Automation

Source Code Management

- Tightly integrated into the IDE
- Built-in support for CVS and Perforce
- Support for day-to-day operations
  - Add, remove files
  - Update, check in changes
  - Compare revisions

Accessing Documentation

- Integrated documentation window
  - API lookup: searches as you type
  - Full-text: search all or part of the documentation
- What's available
  - Xcode help
  - Carbon, Cocoa, IOKit, etc., reference
  - Sun's Java documentation
  - Mac OS X man pages

Scripting and Automation

- Many ways to automate common tasks
  - Drive Xcode using AppleScript
  - Command line build tool
  - User Scripts menu
  - Worksheet functionality

AppleScript Support

- Access details of Xcode project model
  - Change target settings on multiple targets
  - Add files to targets
  - Start and stop builds

Building from the Command Line

- *pbbuild* provides access to IDE build system features
- Can write shell scripts for nightly builds, etc.
- There are switches for controlling:
  - Targets
  - Build styles
  - Overriding build settings

Script Menu

- Completely user-configurable
- Can contain any kind of script
  - AppleScript
  - Shell script
  - Perl, Python, etc.
- Command keys can be bound to scripts
- Scripts can modify contents of text buffers

Worksheet Functionality

- Xcode has MPW-style worksheet functionality
- Can select commands in text buffer and run them
  - Save frequently used code snippets in text files
  - Preserve history indefinitely
- Results are pasted back into the text buffer

Demo

Matt Morse
Tools Engineer, Documentation

Wrap Up

Anders Bertelrud

Summary

- Xcode has many productivity features
- Many work transparently
  - Zero Link, Fix and Continue, etc.
- Others need customization for best results
  - Precompiled headers, scriptability, etc.
- Xcode gives you the power and flexibility

How to Learn More

- Part of Xcode tools package
  - Use the Documentation window and search features
- Inside Mac OS X books
  “Inside Mac OS X: System Overview”
- Apple Developer Connection tools pages
  http://developer.apple.com/tools/
- CVS documentation
  http://www.cvshome.org/docs/
- Perforce documentation
  http://www.perforce.com/perforce/technical.html

Who to Contact

Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

Development Tools Engineering Feedback
xcode-feedback@group.apple.com

To Post Bug Reports
http://developer.apple.com/bugreporter

Mailing Lists at Apple
http://lists.apple.com

http://developer.apple.com/wwdc2003/urls.html

Roadmap

304 GCC In Depth Find out all about advancing a world-class compiler	**The Mission** **Wed., 3:30pm**
305 Tuning with Performance Tools Learn to use the tools and improve performance	**The Presidio** **Wed., 5:00pm**
306 AppleScript Studio Explore professional AppleScript development	**Pacific Heights** **Wed., 5:00pm**
307 Xcode for Carbon Development See how Carbon is supported by Xcode	**Russian Hill** **Thur., 9:00am**
FF009 Dev Tools Feedback Forum Give us your feedback	**North Beach** **Thur., 2:00pm**
308 Debugging Services for Mac OS X Hear the latest in debugging for Mac OS X	**The Presidio** **Thur., 3:30pm**

Q&A

Godfrey DiGiorgi
Technology Manager, Development Tools
ramarren@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference